

LM3S 系列电源管理方式

修订记录

日期	修订	描述	作者
----	----	----	----

目录

1 关键词.....	3
2 处理器控制功耗的方法.....	3
3 时钟频率的功耗测试.....	4
3-1 LM3S系列时钟组件介绍	4
3-2 不同时钟模块工作下的功耗测试.....	5
1 测试内容.....	5
2 测试条件.....	5
3 实验 0: 测试外设功耗.....	6
4 实验 1: 测试PLL模块的功耗.....	6
5 实验 2: 测试INT OSC模块的功耗.....	7
6 实验 3: 测试Main OSC模块的功耗.....	7
7 实验 4: 测试Main OSC+PLL在不同频率下的功耗	8
8 实验 5: 测试 INT OSC下面不同时钟频率的功耗.....	8
4 内核电压的功耗测试.....	9
4-1 LM3S系列LDO组件介绍	9
4-2 不同内核电压工作下的功耗测试.....	9
1 测试内容.....	9
2 实验 1: 测试 50Mhz频率不同内核电压的功耗.....	9
3 实验 2: 测试 8Mhz频率不同内核电压的功耗.....	10
5 工作模式的功耗测试.....	10
5-1 Cortex-M3 对于工作模式的支持.....	10
5-2 LM3S系列工作模式介绍	10
5-2 对LM3S不同工作模式下的功耗测试	11
1 测试内容.....	11
2 测试条件.....	11
3 不同工作频率Sleep模式下的功耗（Disable auto-clock Gating）	11
4 不同工作频率Deep Sleep模式下的功耗（Disable auto-clock Gating）	12
5 不同工作频率Sleep 模式下的功耗（Enable auto-clock Gating）	13
6 Deep Sleep模式下的最低功耗尝试.....	14
6 功耗测试对比总表.....	15

1 关键词

Cortex-M3 系列一个突出的特点就是功耗低。本文以 TI 公司的 LM3S5749 作为例子，通过试验说明 ARM 的 V7M 版本在低功耗方面的独特优势，并对 LM3S5749 的实际功耗情况进行测试，分析不同的节电方式对系统整体性能的影响。

2 处理器控制功耗的方法

本章节主要介绍处理器的控制方式以及 Cortex-M3 系列单片机在功耗控制方面的优势。主要内容如 Table2-1 所示：

节电方式	说明
采用更低的供电电压	IC 器件在正常电压范围内工作电压越低，消耗的电流就越小。这也是很多处理器采用多电源供电的原因，IO 为了符合标准电平标准采用 3.3V 的电压，内核为了节电采用尽可能低的电压。 Cortex-M3 允许更低的内核工作电压。LM3S 系列采用两个电压输入 (3.3V 和 2.1-2.5V) 分别用来为 IO 和内核供电，并且内置 LDO 可以直接把 3.3V 电压转为 2.1V 或者 2.5V 为内核供电。
采用尽可能低的工作频率	对于数字电路，功耗基本上和工作频率成正比。在完成工作的尽可能的采用比较低的工作频率可以降低系统的功耗。 Cortex-M3 的高效设计 (1.25 DMIPS/MHz)，在完成相同的工作需要更低的工作频率。
支持多种工作模式	动态切换工作模式是单片机系统节电的一个重要策略。不同的工作模式 (run、idle、sleep) 需要的电流和可以访问的资源各不相同。 Cortex-M3。Cortex-M3 已经把工作模式上升到了内核设计的高度 (内核级支持睡眠模式和深度睡眠模式) 配合内核级实现的 SysTick 功能可以设计出强大的节电应用程序

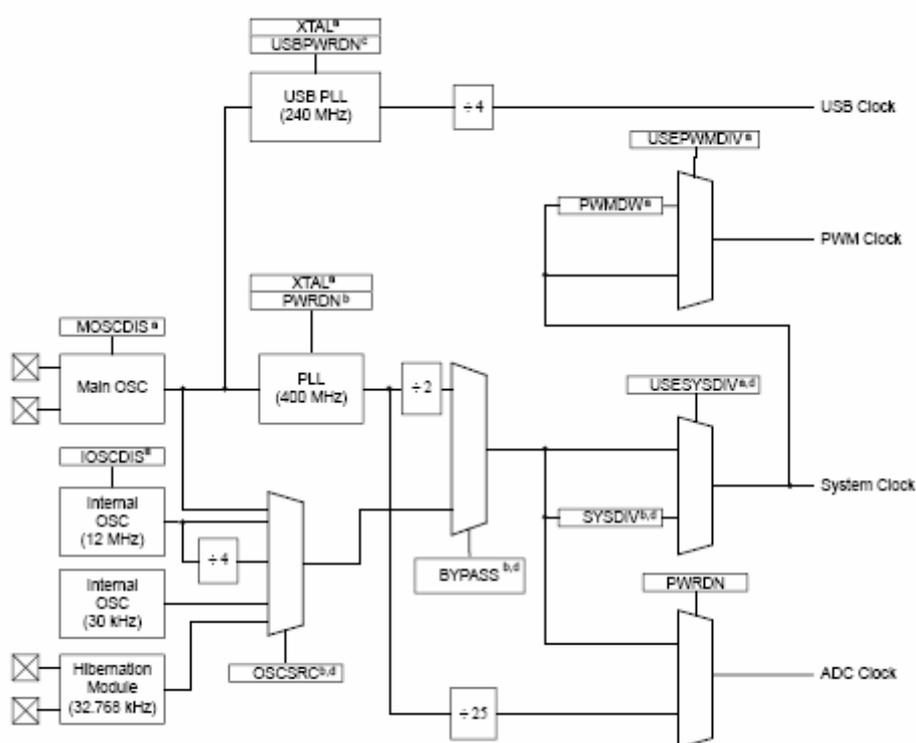
Table2-1

3 时钟频率的功耗测试

本段测试 LM3S 在不同的时钟频率（2MHz-50MHz）及不同的时钟外设下面（内部振荡器、主振荡器、锁相环）的功耗情况。

3-1 LM3S 系列时钟组件介绍

LM3S 系列的时钟组件如图 PIC3.1 所示：



PIC 3.1: LM3S 系列时钟外设

- a. Control provided by RCC register bit/field.
- b. Control provided by RCC register bit/field or RCC2 register bit/field, if overridden with RCC2 register bit USERCC2.
- c. Control provided by RCC2 register bit/field.
- d. Also may be controlled by DSLPCLKCFG when in deep sleep mode.

LM3S 的时钟信号主要有 System Clock, ADC Clock, Pwm Clock, USB Clock 四路。

LM3S 的时钟模块由“Main OSC”“Internal OSC”“PLL”“USB PLL”三部分组成，用这三部分模块为上面提到的四路时钟提供信号。下面分别进行介绍：

Main OSC: 主时钟，需要外接晶振（oscillator）或晶体（Crystal），晶振或晶体的取值范围可以参考 LM3S 的 Datasheet，只有用 Main OSC 才可以配合 PLL 产生 50M 的 System Clock。

Internal Clock: 内部时钟，如果没有晶振或者晶体的时候可以用来产生 System Clock。内部时钟提供一种 30Khz 的模式，在实际设计中可以在任务等待时候切换到内部 30Khz 模式，从而可以大大降低系统功耗。内部时钟的缺点是精度比较低，无法为 USB、UART 等外设定定时。（如果用 USB 或者 UART 就不能用内部时钟）。

PLL: 锁相环，将主时钟的信号进行倍频到 200Mhz。在通过设置合理的分频系数达到 20Mhz 以上的系统时钟（主振荡器不能直接提供 16.384Mhz 以上的时钟信号）

USB PLL: USB 锁相环，主要是为 USB 外设提供稳定的 48Mhz 时钟信号。

3-2 不同时钟模块工作下的功耗测试

1 测试内容

试验主要采用 LM3S 器件提供的驱动程序 SysCtlClockSet ()，通过不同的参数设置得到不同的测试结果。主要完成以下几个实验：

- 1) 实验 0: 测试外设功耗
- 2) 实验 1: 测试PLL模块的功耗
- 3) 实验 2: 测试INT OSC模块的功耗
- 4) 实验 3: 测试Main OSC模块的功耗
- 5) 实验 4: 测试Main OSC+PLL在不同频率下的功耗
- 6) 实验 5: 测试 INT OSC下面不同时钟频率的功耗情况

2 测试条件

核心电压:2.5V

3 实验 0：测试外设功耗

为了验证时钟频率是否正确，需要使能部分 GPIOC 外设。首先测试一下使能的 GPIO 外设的功耗。

- 1) 实验思路：选择同一时钟频率，比较使能外设前后功耗的区别。
- 2) 实验结果

序号	程序及参数	耗电(mA)
实验 1	<code>SysCtlClockSet(SYSCTL_SYSDIV_4 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);</code>	65.55
实验 2	<code>SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC); GPIOPadConfigSet(GPIO_PORTC_BASE,KEY5,GPIO_STRENGTH_2MA, GPIO_PIN_TYPE_STD); GPIODirModeSet(GPIO_PORTC_BASE, KEY5, GPIO_DIR_MODE_IN); GPIOPadConfigSet(GPIO_PORTC_BASE,LED3,GPIO_STRENGTH_2MA, GPIO_PIN_TYPE_STD); SysCtlClockSet(SYSCTL_SYSDIV_4 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ); while (1) { HWREG(GPIO_PORTC_BASE + (GPIO_O_DATA + (LED3 << 2))) = LED3; HWREG(GPIO_PORTC_BASE + (GPIO_O_DATA + (LED3 << 2))) = ~LED3; }</code>	73.2

3) 实验分析

- 验证时钟频率是否正确的 GPIO 在 50Mhz 系统时钟下耗电 $73.2-65.5=7.7\text{mA}$ (注意：不同的系统时钟带来的外设耗电也不相同)
- 验证时钟频率看到的高电平时间长度为 4 个时钟周期(分析下面的汇编程序得到)

C 源程序

```
HWREG(GPIO_PORTC_BASE + (GPIO_O_DATA + (LED3 << 2))) = LED3;
```

对应的汇编程序

```
0x000001FA 2040      MOVS      r0,#0x40  
0x000001FC 4906      LDR      r1,[pc,#24] ; @0x00000218  
0x000001FE 6008      STR      r0,[r1,#0x00]
```

4 实验 1：测试 PLL 模块的功耗

1) 实验思路：

选择同一时钟频率 8Mhz，比较使能 PLL 前后的功耗变化。

2) 实验结果

序号	程序及参数	耗电(mA)	耗电(mA)
		带 GPIO	不带 GPIO
1	SysCtlClockSet(SYSCTL_SYSDIV_25 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	25.57	25.83
2	SysCtlClockSet(SYSCTL_USE_OSC SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	18.35	19.54

3) 实验分析

- 打开 GPIO 总线不一定造成用电增加。(对比上面的两个结果)
- PLL 模块耗电 $25.5-18.3=7\text{mA}$ 左右。

5 实验 2：测试 INT OSC 模块的功耗

1) 实验思路

选择同一时钟频率 8Mhz，比较使能 INT 前后的功耗变化。

2) 实验结果

序号	程序及参数	耗电(mA)
		带 GPIO
1	SysCtlClockSet(SYSCTL_SYSDIV_25 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	18.35
2	SysCtlClockSet(SYSCTL_USE_OSC SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ SYSCTL_INT_OSC_DIS)	17.89

3) 实验分析

- 内部振荡器耗电 0.5mA 左右。

6 实验 3：测试 Main OSC 模块的功耗

1) 实验思路

采用同一频率（12M-内部振荡器），比较使能 Main OSC 前后功耗变化。

2) 实验结果

序号	程序及参数	耗电(mA)带 GPIO
1	SysCtlClockSet(SYSCTL_USE_OSC SYSCTL_OSC_INT);	23.81
2	SysCtlClockSet(SYSCTL_USE_OSC SYSCTL_OSC_INT SYSCTL_MAIN_OSC_DIS);	23.4

3) 实验分析

- 主振荡器耗电 0.4mA 左右。

7 实验 4：测试 Main OSC+PLL 在不同频率下的功耗

1) 实验思路

采用 Main OSC+PLL 产生各种频率的时钟，比较功耗结果。

2) 实验结果

序号	程序及参数	耗电(mA) 不带 GPIO
50	SysCtlClockSet(SYSCTL_SYSDIV_4 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	65.56
25	SysCtlClockSet(SYSCTL_SYSDIV_8 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	41.92
20	SysCtlClockSet(SYSCTL_SYSDIV_10 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	37.24
10	SysCtlClockSet(SYSCTL_SYSDIV_20 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	27.74
8	SysCtlClockSet(SYSCTL_SYSDIV_25 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	25.86

8 实验 5：测试 INT OSC 下面不同时钟频率的功耗

1) 实验思路

采用 INT OSC 产生各种频率的时钟，比较功耗结果

2) 实验结果

序号	程序及参数	耗电(mA) 带 GPIO
1	SysCtlClockSet(SYSCTL_USE_OSC SYSCTL_OSC_INT);	23.81
2	SysCtlClockSet(SYSCTL_USE_OSC SYSCTL_OSC_INT4);	12.87
	SysCtlClockSet(SYSCTL_USE_OSC SYSCTL_OSC_INT30);	9.94

3) 实验分析：

- 当采用 30k 内部时钟以后，依然有 9.94mA 的功耗，功耗来源有待进一步确认。

4 内核电压的功耗测试

4-1 LM3S 系列 LDO 组件介绍

4-2 不同内核电压工作下的功耗测试

1 测试内容

试验主要采用 LM3S 器件提供的驱动程序 SysCtlLDOSet(unsigned long ulVoltage)，通过不同的参数设置得到，主要测试以下电压 2.25、2.50、2.75 下面不同频率的功耗。

2 实验 1：测试 50Mhz 频率不同内核电压的功耗

- 1) 实现思路
- 2) 测试条件
 - SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN | SYSCTL_XTAL_8MHZ);
 - 使能 GPIO 外设/除能 GPIO 外设

3) 测试结果

序号	程序及参数	耗电(mA)	耗电(mA)
		不带 GPIO	带 GPIO
1	SysCtlLDOSet(SYSCTL_LDO_2_25V)	57.5	62.03
2	SysCtlLDOSet(SYSCTL_LDO_2_50V)	66.1	71.06
3	SysCtlLDOSet(SYSCTL_LDO_2_75V)	75.2	80.23

- 4) 实验分析
 - 低 LDO 工作电压的时候如果时钟频率过高，系统不稳定

3 实验 2：测试 8Mhz 频率不同内核电压的功耗

1) 实现思路

2) 测试条件

➤ SysCtlClockSet(SYSCTL_USE_OSC | SYSCTL_OSC_MAIN |
SYSCTL_XTAL_8MHZ);

3) 测试结果

序号	程序及参数	耗电(mA) 不带 GPIO
1	SysCtlLDOSet(SYSCTL_LDO_2_25V)	16.53
2	SysCtlLDOSet(SYSCTL_LDO_2_50V)	19.57
3	SysCtlLDOSet(SYSCTL_LDO_2_75V)	22.88

5 工作模式的功耗测试

5-1 Cortex-M3 对于工作模式的支持

Cortex-M3 已经把工作模式上升到了内核管理的高度，定义了睡眠（sleep）和深度睡眠（deep sleep）两种模式以及 WFI、WFE 两条睡眠指令。在程序中执行 WFI 后者 WFE 进入睡眠模式。

睡眠：由 CM3 微控制器的 SLEEPING 信号指示。

深度睡眠：由 CM3 微控制器的 SLEEPDEEP 信号指示。

注意：SLEEPING 与 SLEEPDEEP 信号都是给芯片的设计者看的。芯片的设计者根据两个不同的信号设计不同等级的节电操作。所以不同品牌的单片机处理睡眠和深度睡眠的操作都不会一致。

睡眠与深睡眠的区分主要根据 NVIC 系统控制寄存器的相关位，关于睡眠指令与睡眠模式的具体介绍可以参考相关书籍。

5-2 LM3S 系列工作模式介绍

LM3S 支持 Sleep ,Deep Sleep, hibernation 三 种模式，Sleep 与 Deep Sleep 是 Cortex-m3 支持的模式，Hibernation 属于 LM3S 的特有功能。

RCC (Run Clock Control)	
RCC2	
RCGC0 (Run Clock Gate Control)	
RCGC1	
RCGC2	
SCGC0 (Sleep Clock Gate Control)	
SCGC1	
SCGC2	
DCGC0 (Deep Sleep Clock Gate Control)	
DCGC1	
DCGC2	
DSLCLKCFG (Deep Sleep Clock Configure)	

5-2 对 LM3S 不同工作模式下的功耗测试

1 测试内容

- 1) 不同工作频率 Sleep 模式下的功耗 (Disable auto-clock Gating)
- 2) 不同工作频率 Deep Sleep 模式下的功耗 (Disable auto-clock Gating)
- 3) 不同工作频率 Sleep 模式下的功耗 (Enable auto-clock Gating)

2 测试条件

核心电压:2.5V

3 不同工作频率 Sleep 模式下的功耗 (Disable auto-clock Gating)

实验思路

采用 Main OSC+PLL 产生各种频率的时钟, 比较功耗进入 Sleep 前后功耗结果。使能 GPIO (PORTC PORTG) 用于显示结果。PORTC 使 LED 闪烁; PORTG 控制按键中断 (按一下进入 sleep 模式, 再按一下唤醒 sleep 模式)。由于 Disable auto-clock Gating, 所以在 sleep 模式下 GPIO 依然会有时钟提供。(RCGC 中相应位在初始化中被设为 1)

```
Void SysCtlPeripheralClockGating(tBoolean bEnable) //开关 auto-clock Gating
```

实验结果

频率	程序及参数	耗电(mA)	耗电(mA)
		Sleep 模式	Run 模式
50	SysCtlClockSet(SYSCTL_SYSDIV_4 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	25	72
25	SysCtlClockSet(SYSCTL_SYSDIV_8 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	18	45
20	SysCtlClockSet(SYSCTL_SYSDIV_10 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	17	40
10	SysCtlClockSet(SYSCTL_SYSDIV_20 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	14	29
8	SysCtlClockSet(SYSCTL_SYSDIV_25 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	13	27
8	SysCtlClockSet(SYSCTL_USE_OSC SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	7	20

实验分析

- 对比 8Mhz 两个实验，验证 PLL 的功耗 7mA 左右
- Sleep 模式的功耗随着工作频率的降低而降低，估计工作频率影响恢复时间
(Datasheet 中没有提到恢复时间相关数据)

4 不同工作频率 Deep Sleep 模式下的功耗 (Disable auto-clock Gating)

实验思路

采用 Main OSC+PLL 产生各种频率的时钟，比较功耗进入 Deep Sleep 前后功耗结果。使能 GPIO (PORTC PORTG) 用于显示结果。PORTC 使 LED 闪烁；PORTG 控制按键中断（按一下进入 Deep sleep 模式，再按一下唤醒 Deep sleep 模式）。由于 Disable auto-clock Gating，所以在 sleep 模式下 GPIO 依然会有时钟提供。

实验结果

频率	程序及参数	耗电(mA)	
		Deep Sleep 模式	Run 模式
50	SysCtlClockSet(SYSCTL_SYSDIV_4 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	5	72
25	SysCtlClockSet(SYSCTL_SYSDIV_8 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	5	45
20	SysCtlClockSet(SYSCTL_SYSDIV_10 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	5	40
10	SysCtlClockSet(SYSCTL_SYSDIV_20 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	5	29
8	SysCtlClockSet(SYSCTL_SYSDIV_25 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	5	27
8	SysCtlClockSet(SYSCTL_USE_OSC SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	7	20

实验分析

- 对比发现使能PLL消耗的电流反而变低，原因见Datasheet P83 (If the PLL is running at the time of the WFI instruction, hardware will power the PLL down and override the SYSDIV field of the active RCC/RCC2 register, to be determined by the DSDIVORIDE setting in the DSLPCLKCFG register, up to /16 or /64 respectively)。初步理解如下：使能PLL的情况下，如果进入Deep Sleep模式，则自动关闭PLL，同时用DSLPCLKCFG寄存器中的相关位（DSDIVORIDE）来代替SYSDIV产生分频系数。也就是说把Main OSC 的实际数值进行DSDIVORIDE分频。DSDIVORIDE缺省值为16，所以Deep Sleep时候系统的时钟等于 $\text{Main OSC (8Mhz) / DSDIVORIDE (16)} = 0.5\text{Mhz}$
- 在除能PLL的情况下，相同的主频，相同的外设，Deep Sleep与Sleep耗电基本相同。

5 不同工作频率 Sleep 模式下的功耗（Enable auto-clock Gating）

试验思路

采用 Main OSC+PLL 产生各种频率的时钟，比较功耗进入 Sleep 前后功耗结果。使能GPIO（PORTC PORTG）用于显示结果。PORTC 使 LED 闪烁；PORTG 控制按键中断（按一下进入 sleep 模式，再按一下唤醒 sleep 模式）。由于 Enable auto-clock Gating，所以在 sleep 模式下 GPIO 不会有时钟提供。（SCGC 中相应位为 0）

```
Void SysCtlPeripheralClockGating(tBoolean bEnable) //开关 auto-clock Gating
```

试验结果

频率	程序及参数	耗电(mA)	
		模式 1	模式 2
50	SysCtlClockSet(SYSCTL_SYSDIV_4 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	22.32	25.28
25	SysCtlClockSet(SYSCTL_SYSDIV_8 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	16.64	18.13
20	SysCtlClockSet(SYSCTL_SYSDIV_10 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	15.53	16.73
10	SysCtlClockSet(SYSCTL_SYSDIV_20 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	13.22	13.82
8	SysCtlClockSet(SYSCTL_SYSDIV_25 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	12.77	13.25
8	SysCtlClockSet(SYSCTL_USE_OSC SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	6.32	6.8

模式 1: Sleep and Enable auto-clock Gating

模式 2: Sleep and Disable auto-clock Gating

试验分析

- 随着频率降低，关闭GPIO带来的功耗降低变得越来越小。
- 在Sleep模式下关闭GPIO时钟会使GPIO的输入中断失效，因为GPIO没有时钟驱动，无法响应中断，所以系统也就无法从Sleep模式下唤醒。

6 Deep Sleep 模式下的最低功耗尝试

实验思路

采用 Main OSC+PLL 产生各种频率的时钟，比较功耗进入 Deep Sleep 前后功耗结果。使能 GPIO (PORTC PORTG) 用于显示结果。PORTC 使 LED 闪烁；PORTG 控制按键中断（按一下进入 Deep sleep 模式，再按一下唤醒 Deep sleep 模式）。在 DSLPCLKCFG 中设置 Deep Sleep 时钟为 INT30，设置分频系数为 64。

实验结果

频率	程序及参数	耗电(mA)	
		Sleep 模式	
50	SysCtlClockSet(SYSCTL_SYSDIV_4 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	5	
25	SysCtlClockSet(SYSCTL_SYSDIV_8 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	5	
20	SysCtlClockSet(SYSCTL_SYSDIV_10 SYSCTL_USE_PLL SYSCTL_OSC_MAIN SYSCTL_XTAL_8MHZ);	5	

	SYSCCTL_OSC_MAIN SYSCCTL_XTAL_8MHZ);		
10	SysCtlClockSet(SYSCCTL_SYSDIV_20 SYSCCTL_USE_PLL SYSCCTL_OSC_MAIN SYSCCTL_XTAL_8MHZ);	5	
8	SysCtlClockSet(SYSCCTL_SYSDIV_25 SYSCCTL_USE_PLL SYSCCTL_OSC_MAIN SYSCCTL_XTAL_8MHZ);	5	
8	SysCtlClockSet(SYSCCTL_USE_OSC SYSCCTL_OSC_MAIN SYSCCTL_XTAL_8MHZ);	7	

实验分析

- 对比发现使能PLL消耗的电流反而变低，原因见Datasheet P83 (If the PLL is running at the time of the WFI instruction, hardware will power the PLL down and override the SYSDIV field of the active RCC/RCC2 register, to be determined by the DSDIVORIDE setting in the DSLPCLKCFG register, up to /16 or /64 respectively)。初步理解如下：使能PLL的情况下，如果进入Deep Sleep模式，则自动关闭PLL，同时用DSLPCLKCFG寄存器中的相关位（DSDIVORIDE）来代替SYSDIV产生分频系数。也就是说把Main OSC 的实际数值进行DSDIVORIDE分频。DSDIVORIDE缺省值为16，所以Deep Sleep时候系统的时钟等于 $\text{Main OSC (8Mhz) / DSDIVORIDE (16) = 0.5Mhz}$
- 在除能PLL的情况下，相同的Main OSC频率，相同的外设，Deep Sleep与Sleep耗电基本相同。

6 功耗测试对比总表

